

50325-0608 (4811)

Patent

UNITED STATES PATENT APPLICATION

FOR

A METHOD FOR CREATING AND DEPLOYING QUALITY OF SERVICE TEMPLATES
IN A PACKET TELEPHONY NETWORK

INVENTORS:

DROR KOREN
GIL SHEINFELD

OF
CISCO SYSTEMS, INC.

PREPARED BY:

HICKMAN PALERMO TRUONG & BECKER LLP
1600 WILLOW STREET
SAN JOSE, CA 95125-5106
(408) 414-1080

EXPRESS MAIL INFORMATION

"Express Mail" mailing label number EL652871446US

Date of Deposit: February 12, 2002

A METHOD FOR CREATING AND DEPLOYING QUALITY OF SERVICE TEMPLATES
IN A PACKET TELEPHONY NETWORK

FIELD OF INVENTION

[0001] The present invention relates generally to configuration management of communication networks, and more specifically to a method for pre-defining and deploying quality of service (“QoS”) policy templates for configuration management of packet telephony networks.

BACKGROUND OF INVENTION

[0002] In today’s high performance internetworks, the need for predictable performance of business-critical applications, as well as the requirements for integrating latency-critical voice and video with bursty data services, mandate differentiated handling of network traffic.

[0003] Quality of Service (“QoS”) enables complex networks to control and predictably service a variety of applications. The goal of QoS is to provide consistent, predictable network services by delivering dedicated bandwidth, controlling jitter and latency, managing congestion and improving traffic flow efficiency. QoS is a set of capabilities that allow the delivery of differentiated services for network traffic, thereby providing better service for selected network traffic. For example, with QoS, bandwidth of critical traffic can be increased, bandwidth for non-critical traffic can be limited, and consistent network response is provided. This allows expensive network connections to be used more efficiently, and service level agreements with customers of the network to be established.

[0004] QoS typically is implemented in software tools that execute in network devices and management stations. QoS tools generally are separated into three categories: classification, queuing and network provisioning. Classification tools mark a packet or flow with a specific priority. This marking establishes a trust boundary that must be enforced. Queuing tools assign a packet or flow to one of several queues, based on classification, for appropriate treatment in the network. Network provisioning tools accurately calculate the required bandwidth needed for voice conversations, all data traffic, any video applications and necessary link management overhead such as routing protocols.

[0005] For effective quality of service treatment, policies must be applied by all intermediate network elements in a path from sender to receiver for a particular message flow. Accordingly, various quality of service management systems are now available. Certain leading systems of this type enable a user or administrator to create and store abstract quality of service policies in terms of collections of rules and policy objects. An example of such a system is QoS Policy Manager, commercially available from Cisco Systems, Inc.

[0006] QoS tools are important in the management of internet protocol (“IP”) telephony networks. Real-time based applications, such as voice applications, have different characteristics and requirements from those of other data applications. Voice applications tolerate minimal variation in the amount of delay affecting delivery of voice packets by routers and other network elements. Voice traffic is also intolerant of lost packets (which causes voice clipping and skips) and unreasonably delayed packets (which can cause either voice quality degradation due to end-to-end voice latency or packet loss if delay is variable). Packet loss, delay and delay variation all contribute to degraded voice quality. QoS tools available for use in packet telephony networks are mechanisms to increase voice quality on data networks by configuring network elements to

result in decreasing dropped voice packets during times of network congestion and by minimizing both the fixed and variable delays encountered in a given voice connection.

[0007] Configuring QoS for voice on an enterprise network is not an easy task. The user needs to know what interfaces to configure and which QoS tools to use on each interface. The user needs to enable QoS not just on a few wide-area network (WAN) links but rather throughout the enterprise, including local-area networks (LAN's). After selecting the QoS tools, user needs to know the command line interface ("CLI") commands translation for use in implementation of configuration policies.

[0008] In the past, many of these tasks have been done manually for a particular device. Design guides, such as the AVVID QoS Design Guide commercially available from Cisco Systems, Inc., provide answers as to what QoS to use and where in order to manually configure QoS across a network. However, the user still needs to telnet to each and every device and set the recommended commands. This is tedious and time-consuming.

[0009] Combining the complexity of enabling QoS for packet telephony with enterprise-wide scaling makes for a daunting, time consuming and costly task. These requirements, along with the susceptibility of human error (e.g., from typographical errors inadvertently included in the configuration) make a strong case for a management solution to make configuring QoS for packet telephony more simple, more scalable and more reliable.

[0010] Based on the foregoing, there is a clear need for a method which will make the process of configuring QoS for voice easier such that the user does not need to know what QoS tools to use, nor do the work of defining all the groups.

[0011] Further, there is a specific need for a method which will enable new and expert users to packet telephony to configure the QoS in their network almost immediately without delving into arcane details underlying QoS techniques and implementations.

00000000000000000000000000000000

SUMMARY OF THE INVENTION

[0012] The foregoing needs, and other needs and objects that will become apparent for the following description, are achieved in the present invention, which comprises in one embodiment, a method for automatically deploying quality of service policies to a plurality of devices in a packet telephony network based on quality of service policy templates.

[0013] In one approach, a method for defining and deploying quality of service policy templates in a packet telephony network is described. A database of network devices and interfaces, including specific values for parameters of each device and interface is provided. A network management system receives location, authentication, and other specific values for devices in the network. The network management system creates and defines a template by receiving information about the network device by launching an SNMP conversation with the device and also a telnet session with the device. These actions allow the network management system to verify that the parameters entered are correct, to detect device and interface information, and to upload QoS policies.

[0014] One feature of this aspect is determining, based on the device QoS information and the interface information, one or more policies that associate traffic flows with the quality of service treatments.

[0015] In another aspect, a method for configuration management of a communications network is provided that enables configuration of several network devices at the same time through the creation of device groups. Templates may reference device groups to accomplish rapid configuration of QoS values on multiple devices.

[0016] In another aspect, a method of storage of a policy template and retrieval of a policy template for implementation on an associated device is provided.

[0017] In another aspect, a method is provided for deploying one or more quality of service policies, as pre-defined in a policy template, to one or more devices in a network. One feature of this aspect is generating a first list of command line interface (“CLI”) commands that correspond to properties for each device. Another feature of this aspect is sending said block of CLI commands to each device to be implemented.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0019] FIG. 1 is a simplified block diagram of an example network context managed by a network management station in which an embodiment may be used.

[0020] FIG. 2A and 2B illustrate example graphical user interfaces in a network management system.

[0021] FIG. 3A and 3B are flow diagrams illustrating a method of creating and defining configuration policies on a packet telephony network.

[0022] FIG. 4 is a block diagram illustrating a network management system from which one or more command line interface (“CLI”) commands based on policies are deployed to one or more network devices in a managed network.

[0023] FIG. 5 is a block diagram that illustrates a computer system upon which an embodiment of the invention may be implemented.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0024] A method for creating and deploying quality of service templates in a packet telephony network is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

[0025] FIG. 1 is a simplified block diagram of an example network context in which an embodiment may be used. The network 100 comprises one or more network devices 102, such as switches, routers, bridges, gateways and other devices. Each network device 102 is coupled to another network device 102 or to one or more end stations 120. Each end station 120 is a terminal node of the network 100 at which some type of work is carried out. For example, an end station is a work station, personal computer, a server or similar device. The network 100 is managed by a network management station 130.

[0026] Each network device 102 executes an operating system 110. An example of such an operating device is the Internetworking Operating System (“IOS”) commercially available from Cisco Systems, Inc. Each network device 102 also executes one or more applications 112 under control of or embedded within the operating system 110. The operating system 110 supervises operation of the applications 112 and communicates with network management station 130 over network connection 104 using an agreed-upon network communication protocol, such as Simple Network Management Protocol (“SNMP”).

[0027] A quality of service policy 10 is defined using network management system 130 and is deployed to a network device 102. The quality of service policy 10 may be defined by a user of a quality of service network management system. Further, the QoS policy 10 may be defined in one or more QoS policy templates 12 stored in a database 115 for implementation across the network.

[0028] Quality of service policies specify, in an abstract manner, what quality of service treatment should be applied to a particular type of network traffic. A QoS policy is a conditional statement that applies one or more specified QoS actions to a packet if the packet satisfies the conditions (filters) defined in the policy. An example of a policy definition is: “ERTraffic,” “Protocol is TCP and Source is ERPServer,” “Coloring,Precedence=Critical(5)”. This policy means “for traffic associated with an Enterprise Resource Planning (“ERP”) application when the protocol is TCP and the traffic originates from the ERPServer, apply packet coloring with a precedence value of Critical.” Policies may call for packet coloring, traffic shaping, limiting actions on routers or switches, priority queuing, custom queuing, etc.

[0029] As illustrated in FIG. 1, a network management system 130, such as Cisco QoS Policy Manager 2.1 commercially available from Cisco Systems, Inc., continually monitors the network and maintains a database 115 of information for every managed device in the network. According to one embodiment of this invention, a configuration manager within network management system 130 obtains the values of certain characteristics which define the characteristics of the device being managed by querying the device. The configuration manager then enables a system administrator, via a user interface, to use this information to configure and manage the device in the network community. For example, the administrator may create new

configurations, load these configurations to devices anywhere on the network and then verify whether the configurations have changed.

[0030] FIG. 2A illustrates an example graphical user interface in a network management system. The GUI of FIG. 2A may be generated under software control by network management system 130, for example.

[0031] In interface 200, a main network management system window 202 is divided into three panes 204, 206, 208. A tree view 210 is located in a tree view pane 204, which displays a hierarchical view of the devices and device groups being managed, and their associated interfaces. For example, a Devices folder 212 contains a separate sub-folder or object 212A-212N for each device. A Device Groups directory 214 contains device groups defined by user. A list view 216 is located in a list view pane 206, which displays the policies defined for the interface, device or device group selected in the tree view pane, if any. A properties preview 218 is located in a properties preview pane 208 that displays the properties of a device, interface or policy selected in the tree or list view panes. Information in this pane is useful in verifying that properties and filter conditions have been defined properly.

[0032] FIG. 2B is a diagram of a tree view of the kind shown in FIG. 2A. Each tree view 210 is rooted at a database node 220 having a name corresponding to a database that is in database 115 of FIG. 1. Thus, there may be multiple sub-trees associated with different template databases within database 115. In the example of FIG. 2B, a tree view of a "Tutorial" database is displayed.

[0033] Each database node 220 has, as immediate child nodes, a Devices folder 212 and a Device Groups folder 214. Devices folder 212 has child nodes for one or more devices 212A, 212B, 212N. In one embodiment, devices 212A, 212B, 212N are identified within tree view 210

by IP addresses associated with physical network devices that are represented by 212A, 212B, 212N. Thus, in the example of FIG. 2B, device 212A is named “10.10.10.11,” device 212B is named “10.2.2.2,” etc. Each device 212A has child nodes for each of its interfaces. For example, device 212B has a first interface 222A designated “Ethernet2/0” and a second interface 222B designated “Serial3/0.”

[0034] Device Groups folder 214 contains child nodes that identify devices and interfaces that are grouped in a meaningful way. For example, device group 214A, designated “EdgeGroupInbound,” contains first and second interfaces 224A, 224B named “10.2.2.2\Enternet 2/0” and “10.4.4.4\Enternet2/0.” Interfaces identified in a particular group may physically exist in different devices. Thus, first interface 224A is on device 10.2.2.2 whereas second interface 224B is on device 10.4.4.4.

[0035] In the approach herein, a user is able to configure QoS values and parameters for all devices in the network using one or more QoS templates. Each template lists all the properties specific to one or more devices or device groups for the most effective QoS for packet telephony networks. A template is created based on information about the device and its interfaces in the database.

[0036] FIG. 3A and 3B are flow diagrams illustrating a method of creating and defining configuration policies on a packet telephony network. In block 302 of FIG 3A, the QoS requirements for a configuration solution are analyzed, and the points on the network where QoS is required are determined. This may be carried out using manual analysis of traffic requirements, such as the amount of latency acceptable in a link, the amount of acceptable jitter, etc. Determining points on the network for enforcing QoS may be carried out by analyzing a

network topology display using a conventional network management software program. The Design Guide referenced above may be used for assistance.

[0037] In block 304, for each network point, all the possible devices and network elements are found. This may be carried out by issuing queries to an inventory database of a network management system. The Design Guide referenced above may be used for assistance.

[0038] In block 306, for each network element, specific QoS tools to be used are defined. This may be carried out by selecting one or more QoS tools from a menu of available QoS tools using the user interface of a software application program.

[0039] Information about the device and its interface parameters are received by the network management system database before policies or templates are created

[0040] In block 308 of FIG. 3B, the network management system receives device location and authentication information from the user. In one embodiment, the following information is supplied by the user through a user interface to the network management system: 1) the IP address for the device, 2) the SNMP read community string for the device, 3) the password required for Telnet access to the device and 4) the password required to enter “enable” mode on the device.

[0041] In block 310, information identifying the device’s interfaces is received. Using the information received in block 308, the network management system launches an SNMP conversation or telnet session with the device. Device data can also be taken from virtual devices that are created as text or XML files, etc. These allow the network management system to verify that the parameters entered are correct and to detect device information in order to upload QoS policies within a template.

[0042] Further, the templates can be defined without real devices. The templates have device parameters in their description, but these parameters can be entered manually and such entry does not require live communication with real devices.

[0043] If the network device is online, the network management system queries the device, determines the device model and software version and obtains a list of the device's interfaces. The network management system then creates a folder for the device in the tree view 210 (FIG. 2) using the IP address of the device and the interface information. If the device is not online, then the device and its interfaces may be added manually by entering the parameters via a user interface. The device is then included in the tree view with its interfaces listed below.

[0044] Device groups in a template may be utilized to allow a user to treat selected interfaces or subinterfaces as a single unit, so that common policies or QoS settings can be easily applied to the group across the network. For example, access ports for a specific device may require the same uniform set of QoS policies throughout the enterprise. Rather than setting these policies port by port or even machine by machine, a template can associate all such related policies with a single device group sharing a single centralized QoS policy.

[0045] In block 312, a device group is created in which the device and its interfaces are added to the database 115. Thereafter, the network management system 130 receives from the user a meaningful name for the device group, device model, software version, interface type, card type, and what characteristic the group contains in common. Device groups are an efficient way of administering QoS policies because of their ability to centralize and scale policies. In addition, change management is easier using device groups.

[0046] In block 314, a policy is created for the devices in each device group based on information received by the network management system during its SNMP and telnet queries

with the device. The policy is created by user interaction with network management system 130. When a policy is created, the network management system only presents to the user those actions and settings that are valid for the device and interface as they are defined in the QoS database. The actions that can be applied using QoS policies depend on several factors, such as the QoS property defined on the interface or device group, the type of device, the type of interface, the type of card, the software version for the device, etc.

[0047] In order to create a new network configuration policy such as coloring treatments, queuing, etc., a user selects a device or interface. The user gives the policy a meaningful name. The direction of the traffic flow is determined, (i.e., whether inbound or outbound) and the policy's filter is created. The filters determine to which packets the policy applies. The filters created for a policy can be broad, in which case the policy is applied to a high percentage of traffic that travels through the device or interface, or it can be narrow and selective. The filter elements available to the user differ depending on the type of device. Typically, the traffic is identified by its general characteristics, its source or destination.

[0048] The filter applies precedence to those packets that satisfy the filter criteria. A user may select to give precedence or higher priority to specific traffic, such as voice traffic. When the device determines that a packet satisfies the conditions of the policy, it applies the policy's action to it. If there is more than one policy defined on the interface or device, the device examines the policies in order, top to bottom, until a match is found, at which point it applies the policy and ignores remaining policies.

[0049] In block 316, the policy's action is defined. Within a packet telephony network, different policy actions can be taken to provide network-wide QoS to expedite the transmission of voice packets while reducing jitter. These mechanisms include coloring, shaping, limiting and

queueing. Coloring is used to change the logical color value associated with a packet, thus changing the packet's relative importance. Shaping is used to smooth the rate of an outbound traffic flow. Shaping policies buffer traffic flows to even out the transmission rate; packets are not dropped until the buffer limits are reached. Limiting is used to limit the bandwidth available to traffic. Queuing may be used to assign outbound traffic to a priority queue or for congestion management to ensure that higher priority packets are given weighted precedence over lower priority packets. The following are examples of features that can also be used to manage voice traffic: low-latency queuing ("LLQ"), strict priority queuing, frame relay fragmentation, link fragmentation and interleaving, and compressed real-time protocol.

[0050] In block 318, the policy that has been created is stored in a policy template. Provisioning network resources based on the relative importance of application traffic is an effective way to deliver QoS in packet telephony networks. Packet classification allows the appropriate packets traversing a network element or particular interface to be selected for QoS service. Using policy based templates, a user can quickly construct rules-based QoS policies that identify and partition traffic into multiple classes of service. When classified, these packets can then be marked for the appropriate IP precedence or differentiated services code point ("DSCP").

In block 320, device groups are associated with interfaces. In this stage, the device group is ready with parameters and policies and is associated with the interfaces brought into block 310.

[0051] Each resulting policy template contains values collected from the device associated with QoS policies specific to the device for effective traffic management in packet telephony networks. The template may be stored in a centralized, network-wide policy database, such as database 115, or another storage device. The database 115 maintains a knowledge base that stores attribute information on the QoS capabilities of each device. This knowledge base ensures

that QoS policies are defined only for QoS mechanisms supported by target devices and then translates these QoS policies into configuration commands specific to each interface. This policy abstraction and automation reduces repetitive tasks associated with defining policies for multiple devices, ensuring QoS policy integrity and reducing the cost and time involved in deploying complex policies to numerous devices.

[0052] Table 1 presents an example template for providing a Cisco Catalyst 6000 switch with access to IP phones:

TABLE 1—EXAMPLE TEMPLATE FOR CATALYST 6000 DEVICE GROUP

Device Group Name: Acc6000=>IP-Phones

Properties of device group:

Model: Cat6000

Mapped Software Version: 5.5

Group Contains Interfaces, Type: Ethernet, Card Type:
Non-VIP

QoS Property of Device Group “ Acc6000=>IP-Phones “:
2Q2T/1P2Q2T

Trust State: Trust QoS, Trust-ext: Untrusted, QoS Style:
VLAN Based

CLI commands:

```
cat6k-access> (enable) set port qos <ip-phone-port(s)> trust-ext  
untrusted
```

```
cat6k-access> (enable) set port qos <ip-phone-port(s)> trust trust-qos  
cat6k-access> (enable) set port qos <ip-phone-port(s)> vlan-based
```

[0053] Each template comprises a name, a set of device group properties, and a set of associated CLI commands. The template name provides a meaningful name that suggests what QoS properties are applied to what types of devices, when the template is invoked. The set of device group properties identifies attributes that must be found in a particular device in order to apply the template to that device. In the example of Table 1, device group properties include the device model, the mapped software version, the interface type, the card type, the common QoS

property of the device group, the trust states and QoS style of the group. The CLI commands identify commands that are issued to devices having the device group properties when the template is invoked. When executed by the device, the CLI commands set QoS policies for the device.

[0054] In one embodiment, when a user selects a template by name from within tree view pane 204 in the user interface of FIG. 2A, the device group properties and CLI commands associated with the template are automatically displayed in Properties preview pane 208, and a list of devices in the network that match the template are displayed in List view pane 206. This enables a user to rapidly see, in real time, what commands will be deployed to what devices to result in implementing the QoS policy suggested by the name of the template.

[0055] After policies have been saved in the network management database, they are deployed, as shown by block 322 of FIG. 3. Deployment refers to distributing policies to the network devices so they can be implemented. A distribution manager may be used to distribute invoke the templates and distribute the device configurations and QoS policies from database 115 to the network 100 and device 102. The network management system translates the policies into device commands, based on the templates, and sends the commands to the devices through the command line interface (“CLI”). Thus, to deploy a quality of service policy to one or more devices in a network, the policy management system converts each policy into one or more CLI commands that the operating system of the device can process. When a device executes the CLI commands of a new configuration, the device becomes reconfigured and will implement the quality of service policy as the device handles network traffic.

[0056] FIG. 4 is a block diagram of a network management system from which policies are deployed. A network management system 130 includes, among other elements, one or more

stored quality of service policies 10 and CLI commands 402. One or more CLI commands 402, based on the policies 10, are deployed to one or more network devices 102 in a managed network 100.

[0057] Network device 102 is one or more switches, routers, gateways, etc. Network 100 is one or more local area networks, wide area networks, campus networks, internetworks, etc. QPM-Pro, commercially available by Cisco Systems, Inc., creates a distribution job based on the policy definitions in the selected database. This job consists of the commands required to reconfigure the devices to implement your policies. The list view will show the devices whose configurations will be changed by the job.

[0058] Through the network management system, commands that will be used to configure the devices can be inspected. During the policy distribution, the device log messages can be viewed as the network management system configures each device, so that successes and failures can be identified.

[0059] FIG. 5 is a block diagram that illustrates a computer system 500 upon which an embodiment of the invention may be implemented. Computer system 500 includes a bus 502 or other communication mechanism for communicating information, and a processor 504 coupled with bus 502 for processing information. Computer system 500 also includes a main memory 506, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 502 for storing information and instructions to be executed by processor 504. Main memory 506 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 504. Computer system 500 further includes a read only memory (ROM) 508 or other static storage device coupled to bus 502 for storing static information and instructions for processor 504. A storage device 510, such as a

magnetic disk or optical disk, is provided and coupled to bus 502 for storing information and instructions.

[0060] Computer system 500 may be coupled via bus 502 to a display 512, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 514, including alphanumeric and other keys, is coupled to bus 502 for communicating information and command selections to processor 504. Another type of user input device is cursor control 516, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 504 and for controlling cursor movement on display 512. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0061] The invention is related to the use of computer system 500 for defining and deploying quality of service policy templates. According to one embodiment of the invention, defining and deploying quality of service policy templates is provided by computer system 500 in response to processor 504 executing one or more sequences of one or more instructions contained in main memory 506. Such instructions may be read into main memory 506 from another computer-readable medium, such as storage device 510. Execution of the sequences of instructions contained in main memory 506 causes processor 504 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

[0062] The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 504 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission

media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 510. Volatile media includes dynamic memory, such as main memory 506.

Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 502. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0063] Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0064] Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 504 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 500 can receive the data on the telephone line and use an infrared transmitter to convert the data to an infrared signal. An infrared detector can receive the data carried in the infrared signal and appropriate circuitry can place the data on bus 502. Bus 502 carries the data to main memory 506, from which processor 504 retrieves and executes the instructions. The instructions received by main memory 506 may optionally be stored on storage device 510 either before or after execution by processor 504.

[0065] Computer system 500 also includes a communication interface 518 coupled to bus 502. Communication interface 518 provides a two-way data communication coupling to a network link 520 that is connected to a local network 522. For example, communication interface 518 may be an

integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 518 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 518 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0066] Network link 520 typically provides data communication through one or more networks to other data devices. For example, network link 520 may provide a connection through local network 522 to a host computer 524 or to data equipment operated by an Internet Service Provider (ISP) 526. ISP 526 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 528. Local network 522 and Internet 528 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 520 and through communication interface 518, which carry the digital data to and from computer system 500, are exemplary forms of carrier waves transporting the information.

[0067] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.